SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 0000; **00**:1–24
DOI: 10.1002/sec

RESEARCH ARTICLE

Flashing displays: User-friendly solution for bootstrapping secure associations between multiple constrained wireless devices

Tonko Kovačević[1], Toni Perković[2]* and Mario Čagalj[2].

[1]University Department of Professional Studies, University of Split, Croatia [2]Department of Electrical Engineering, FESB, University of Split, Croatia

ABSTRACT

Future healthcare systems, smart homes, and similar will involve a large number of *smart* inter-connected wireless devices (such as wireless sensor nodes). One of the major challenges to securing these systems presents loading initial cryptographic credentials into a relatively large number of wireless devices. Furthermore, many of these technologies involve low-cost and highly interface constrained devices (lacking usual wired interfaces, displays, keypads, and alike). We propose two novel multichannel key deployment schemes for wireless networks that only require a presence of a light source device, such as a multi-touch screen (tablet or smartphone device). The *first* key deployment scheme is based on secret key cryptography and is suitable for interface/resource constrained wireless devices. The second scheme assumes a strong attacker and requires the use of public key cryptography. In both our solutions we use one-way visible light channel (VLC) of multi-touch screens (flashing displays) to initialize devices in a secure, usable and scalable way. From the user's perspective, this boils down to placing the devices on the multitouch screen after which the remaining process is fully automatized. We showed through the experiments with 48 users that our solution is user-friendly and scales linearly with the number of nodes.

KEYWORDS

Wireless devices, key deployment, visible light channel, authentication, signal cancellation, interference.

***Correspondence**

Department of Electrical Engineering, FESB, University of Split, Croatia. E-mail: toperkov@fesb.hr

Received . . .

1. INTRODUCTION

The Internet of Things (IoT) is a new ubiquitous-computing paradigm that seeks to enhance the traditional Internet by creating intelligent interconnections of diverse objects in the physical world. This network generally encompasses a large number of wireless devices that lack traditional user interfaces (like keyboards, keypads, displays), and often have limited computing and energy resources (e.g., tiny wireless sensor devices). IoT applications range from Smart Homes, e/m-healthcare systems, Smart Cities, Intelligent Transport Systems, etc. In a typical m-healthcare scenario, a user (being a nurse, a physician or a patient) would like to setup an ad hoc network comprising a set of small medical wireless sensing devices such as a thermometer, a heart rate and/or blood oxygen meter, or some other medical or general-purpose sensing device. These sensing devices are capable of communicating with each other and with personal gadgets such as smartphones, tablets, smart TVs, etc., over short range wireless technologies such as Bluetooth and WiFi. Moreover, the sensing devices will be able to exchange data with remote web/cloud servers either directly, using for example GSM/UMTS technology, or indirectly using different WiFi proxies (home access points, smartphone-based hotspots, etc.). Of course, prior to any communication taking place, the user would like to take appropriate steps towards ensuring privacy and integrity of information of personal nature (e.g., personal data readings). However, the problem of bootstrapping a secure communication between such wireless devices (a.k.a. - *secure network bootstrapping/initialization*) presents a great challenge [1, 2], especially for devices, such as iBeacons [3, 4] or LIFX smart bulbs [5], that lack traditional rich user interfaces (keyboards, keypads and displays). Ideally, establishing secure associations between interface limited devices and smartphones/tablets, between interface limited devices and remote cloud servers, and between the inteface limited devices themselves, should be very easy and natural for the user to carry out, without requiring him/her to go through complex and error prone device configuration procedures (involving extra cables, selecting and typing passwords in each and every network device, reading long instruction manuals). In this work, we focus specifically on the problem of establishing initial security associations between a potentially larger group of interface constrained wireless devices that do not share any authentic information, like secret passwords, pins, keys, certificates or similar common knowledge, in advance. Our ultimate goal is to secure communication within the given group of devices, i.e., ensure that each group member can establish an authentic and private channel with any other group member. We refer to this problem as to *secure network bootstrapping* or *secure network initialization*. Although in this work we do not explicitly address the problem of securing communication outside the given group of wireless devices (e.g., communication between a group member and a remote server), the solutions we develop here for bootstrapping secure within-group communication can easily be adapted for this purpose.

Many existing proposals for secure network bootstrapping assume that the network nodes already share a secret key (preloaded at the manufacturing time) that can be used to bootstrap secure key agreement at later stages [6, 7, 8].

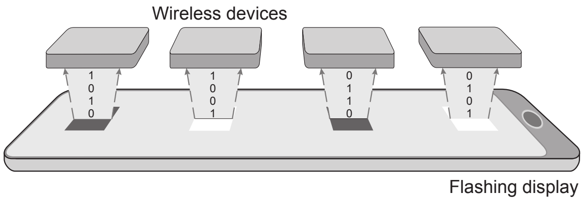


Figure 1. Blinking screen. Transmission of messages over a visible light channel (VLC).

However, a recent weakness found in LIFX smart bulbs [5] shows that this approach comes with a great risk: the compromise of a single device (the preloaded shared secret key) can lead to the compromise of the whole network or even the whole line of devices. The case with LIFX smart bulbs only confirms the fact that users may not always trust the keys preloaded by the manufacturer. Some other solutions for secure network bootstrapping propose sending the keys in clear over an insecure wireless channel, assuming that the attacker will not be present during the key deployment [9]. Another line of work, including Message-in-a-Bottle [10] and KALWEN [11], rely on the availability of a specialized setup hardware (e.g., a Faraday cage) during the key deployment. While very secure, the need for specialized hardware makes the solution quite expensive and difficult to use. Some other well-established secure bootstrapping solutions are based on *multichannel protocols* [12] where communication between network devices takes place over two channels, an insecure high bandwidth radio channel and a special low bandwidth *out-of-band (OoB)* channel, such as visible light (a.k.a. - *visible light channel/communication (VLC)*) or acoustic channel. The solutions presented in [13, 14, 15] are based on the multichannel approach that involve a camera and visible light communication. This approach scales very well with the number of devices in the network, but can still be somewhat involved for the end user who has to setup and position the camera. Perkovic et al. [16] and Li et. al. [17] also propose solutions which exploit an auxiliary OoB visible light channel, but require no additional specialized hardware apart from the network devices themselves. While completely eliminating the need for extra/specialized hardware, these solutions can be very demanding for an end user in some of their aspects (see Perkovic et al. [16] for more details). Similar solutions (e.g., GAnGs [18] and Groupthink [19]) are developed for multi-user setting where each user has a personal device,

DOI: 10.1002/sec
Prepared using secauth.cls

such as a smartphone. Unfortunately, these solutions are not directly applicable to our setting as they require interfaces such as displays, keyboards and/or cameras.

1.1. Requirements for secure initialization

Building on the large body of existing work, it is our goal in this paper to develop a user-friendly and scalable mechanism for secure bootstrapping of a larger network of interface constrained wireless devices, without any prerequisite for shared secrets or other form of authentic common knowledge (such as passwords, secret keys and certificates) among the network devices. The bootstrapping mechanism should also satisfy the following requirements:

1. *User-friendliness*: The network bootstrapping mechanism should be very simple, intuitive, and easily administered by a non-specialist and an unaided end user.
2. *Scalability*: Users should be able to initialize a reasonably large number of wireless devices. Due to the size of initialization setup (screen size as well as the device size), however, one can only hope to initialize a maximum of, for example, 10-20 devices per batch of initialization.
3. *Compatibility with resource and interface constrained devices*: The devices can have limited energy, memory and computational power and may lack rich user interfaces, such as displays and/or keypads. In other words, a secure network bootstrapping should be possible with minimal hardware requirements (e.g., a photodiode (or a LED) and a push-button[*]).

1.2. Our contributions

In this paper, we present two *multichannel* schemes for secure bootstrapping of a large group of interface constrained wireless devices into a network. Both our schemes involve communication over a regular high-bandwidth radio channel and an out-of-band *visible light communication (VLC)* channel. Out-of-band visible light communication is implemented using a flashing display whose flashing lights are received by the interface constrained devices using their light sensors (cheap photodiodes), as depicted in Fig. 1. The first proposed network bootstrapping scheme, named LIRA, is based on secret key cryptography, and therefore it is suitable for highly CPU-constrained devices/nodes. LIRA has been initially proposed in [21], and here we extend it in several important ways: we simplify and optimize the original LIRA protocol (by reducing the number of messages required for secure bootstrapping), we formally verify the security properties of LIRA protocol using Scyther tool[†], and finally, we substantially extend the usability study related to LIRA from [21]. In LIRA protocol, a light source (LS) unit (a screen of a tablet, smartphone, laptop, etc.) transmits secret keys over a protected visible light channel to a group of wireless devices (Fig. 1). In the second phase of the protocol, one device takes the role of a group coordinator and runs a key verification process with the remaining group members over an insecure radio channel. Light is easier to block and hide from an eavesdropping adversary than radio waves and that makes our approach easier and less cumbersome to secure than, e.g., Message-in-a-bottle [10] and KALWEN [11]. Moreover, since any source of light, e.g., a laptop, a smartphone, or a tablet screen, can be used to transmit the keys, many nodes can be programmed at the same time without the need for specialized hardware, increasing usability and overall speed. We implemented LIRA protocol on commercially available (interface constrained) wireless platforms and we show the performance and scalability of our system through extensive usability experiments with 48 users. The results of the study indicate that LIRA protocol is easy to use, it is robust to user errors and it achieves fast protocol execution times.

Although light signal is harder to intercept than a regular radio signal, it would still be possible to eavesdrop data (secret keys in our LIRA protocol) transmitted using a flashing screen with collocated suitably sensitive photo-detectors (e.g., a regular camera). For example, a commercial solution BlinkUp developed by Electric Imp [22] is vulnerable against such an adversary. A more dedicated adversary could also observe the electromagnetic radiation (from faraway and non line-of-sight locations) generated by a flashing screen [23], thus potentially compromising the content (secret keys) shown on the screen. Therefore, our second mechanism for secure network bootstrapping employs public key cryptography

[*] Large number of commercially available devices contain LED and a button [20].

[†] http://www.cs.ox.ac.uk/people/cas.cremers/scyther

DOI: 10.1002/sec
Prepared using secauth.cls

and assumes a much stronger adversary who is capable of reading the contents of the flashing screen at any moment. In this scheme, we use VLC channel in combination with a security primitive *integrity codes (I-codes)* [24]. Unlike similar approaches in [25, 16], where VLC is used for transmission of short authentication strings, in the present scheme VLC is used only as a means to synchronize specially crafted (I-coded) radio messages and to easy the process of loading the group size info into interface limited devices. Similar approaches based on I-codes appear in [26] (TEP protocol) and in [27] (Chorus protocol). However, TEP is a pairing protocol intended for two devices only, while Chorus require users to manually enter the group size into at least one legitimate device[‡], which can be very demanding (error-prone) to accomplish on devices with limited user interfaces [16, 28]. On the contrary, our solutions utilizing a multitouch screen as the light source, scale well to hundreds of devices and make the problem of counting and informing the devices about the size of the initialization group rather trivial.

Finally, we also add to better understanding of security and robustness of I-code security primitive [24], by studying its behavior in realistic (low and high interference) environments both analytically and through experiments.

1.3. Motivation to utilize touchscreens

Today, touchscreens are ubiquitous in our everyday lives be it on smartphones, tablets, laptopts, desktop monitors, different appliances, they can be found just everywhere. As mentioned above, a common requirement by many group authentication protocols that support groups of an arbitrary size (two or more devices/entities) is that at least one device from that group knows the correct group size. To meet this requirement, many such protocols require users to manually enter the group size into at least one legitimate device. While this is easy to do on devices equipped with keypads, keyboards or verify on displays, it is especially challenging to carry out the same task on devices that have very constrained interfaces (e.g., have a single push-button and a LED) [16, 28]. Being omnipresent touchscreens are an excellent choice for accomplishing this goal as no specialized setup hardware nor specialized device hardware is required; a touchscreen as a VLC transmitter and a simple and cheap photodiode on the device side suffice. Multitouch screens can be further used to greatly ease and simplify different network administration tasks when deploying a network of resource/interface constrained devices. Thus, tasks such as assigning different sensing devices to different roles (e.g., a room sensor, a kitchen sensor), grouping devices and establishing some logical hierarchy between them, could all be accomplished by simply circling the devices placed on the touchscreen and/or connecting their positions on the screen by lines. This is a very intuitive approach, very robust to user errors, and the approach that puts the user at the center. We do not explore this line of research further in the present work, we leave it as a potential future work.

The rest of this paper is organized as follows: in Section 2 we describe the symmetric key protocols, and the security analysis of these protocols is given in Section 3. The public key based protocol and and its security analysis is given in Section 4. Usability evaluation and related work are provided in Section 5 and 6. Finally, we conclude in Section 7.

2. NETWORK BOOTSTRAPPING BASED ON SECRET KEY CRYPTOGRAPHY

In this section we describe LIRA (*Light channel for device Initialization and Radio channel for Authentication*), a secret key cryptography-based multichannel protocol for secure bootstrapping of a network of interface constrained wireless devices. In LIRA, we initially use a private one-way visible light channel to transfer secret keys to the network devices, then the radio channel is used to confirm and verify their correct reception (the deployed keys can later be used for mutual authentication between the group/network devices as well as for establishing private communication channels).

2.1. Key transmission over a visible light channel

LIRA protocol for secure network bootstrapping is based on adapted ISO/IEC 9798-4 [29] three-pass mutual authentication protocol, as shown in Fig. 2. We

[‡] This is essential for the security of such protocols, since otherwise an unauthorized (malicious) device could easily join the group/network as a legitimate member.

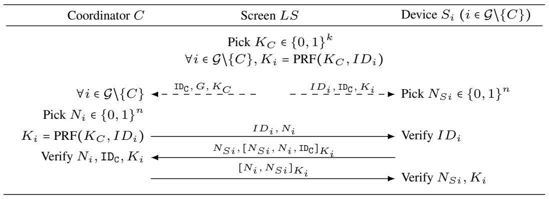

Coordinator C
Screen LS
Device S_i $(i \in \mathcal{G}\backslash\{C\})$
Pick $K_C \in \{0,1\}^k$
$\forall i \in \mathcal{G}\backslash\{C\}, K_i = \mathrm{PRF}(K_C, ID_i)$
$\forall i \in \mathcal{G}\backslash\{C\}$
$\mathtt{ID_C}, G, K_C$
$ID_i, \mathtt{ID_C}, K_i$
Pick $N_{Si} \in \{0,1\}^n$
Pick $N_i \in \{0,1\}^n$
$K_i = \mathrm{PRF}(K_C, ID_i)$
ID_i, N_i
Verify ID_i
$N_{Si}, [N_{Si}, N_i, \mathtt{ID_C}]_{K_i}$
Verify $N_i, \mathtt{ID_C}, K_i$
$[N_i, N_{Si}]_{K_i}$
Verify N_{Si}, K_i

Figure 2. Multichannel LIRA protocol: Bootstrapping a group ($\mathcal{G}$) of $G = |\mathcal{G}|$ interface constrained wireless devices; the dashed arrows indicate communication over a private *visible light channel*, and the solid arrows represent communication over a public radio channel.

modified the original ISO/IEC 9798-4-4 [29] protocol so to incorporate a one-way visible light communication (VLC) channel. Secure network bootstrapping with LIRA protocol works in the following way. A user wishes to enable secure communication between a group $\mathcal{G}$ of interface limited wireless devices. We denote with G the size of the group, i.e., $G = |\mathcal{G}|$, and we assume the devices from $\mathcal{G}$ to be trusted. At the onset of the bootstrapping process, the user designates one arbitrary device from $\mathcal{G}$ as a coordinator, represented by C. Depending on the setting, the coordinator can be a specialized device (e.g., serving the role of an internet proxy) or just an ordinary device (one among equals). To accomplish this task, the user simply places the intended coordinator device on the touchscreen, i.e., the light source unit (LS), as the first device from $\mathcal{G}$, so that the screen can learn its position. Alternatively, the user could place the intended coordinator device at the predefined spot on the screen, or simply circle it once all the devices from $\mathcal{G}$ have been placed on the screen. The application logic behind the role of the LIRA light source unit can be implemented as a standalone application or even better as a web application[§].

The user then places the remaining devices from the group, represented by S_i, $i \in \mathcal{G}$, on the touchscreen and powers them on. Each device should be placed in such a way that it can measure the light intensity of the flashing fragment of the screen that corresponds to the device's location. Moreover, the device should completely block the light coming from the flashing part of the screen below it (later in Section 4 we relax this requirement). At the same time the touchscreen (LS unit) determines the position of each device placed on it and their total number, i.e., G.

Following LIRA protocol shown in Fig. 2, at this stage of the protocol the light source unit LS allocates a unique session identity ID_i to each device $i \in \mathcal{G}$. The coordinator C is allocated a special fixed identity, represented by a string $\mathtt{ID_C}$, which is known apriori to all the protocol entities; $\mathtt{ID_C}$ is essentially a public and system-wide parameter. In LIRA, the coordinator C is always allocated the number 0 as its unique identity ($\mathtt{ID_C} = 0$), while the remaining devices are allocated sequentially increasing identities, starting with 1 and ending with $G - 1$. LS unit also generates a random k-bit *master secret key* K_C to be used by the coordinator. After that, a *pseudo-random function* PRF (e.g., HMAC) is applied to the allocated IDs in order to derive secret keys for the remaining devices as follows: $K_i = \mathrm{PRF}(K_C, ID_i)$, $\forall i \in \mathcal{G}\backslash\{C\}$.

Referring to Fig. 2 (the dashed arrows), at this stage LS sends simultaneously the following message flow $\{ID_i, \mathtt{ID_C}, K_i\}$, over a private one-way visible light channel, to each device S_i, $i \in \mathcal{G}\backslash\{C\}$. At the same time, LS sends the message flow $\{\mathtt{ID_C}, G, K_C\}$ to the coordinator device. Upon the reception of its message flow, the device designated as a coordinator learns its role, the number of devices in the group (i.e., G), and the master secret key K_C. Similarly, the remaining devices learn their own role, the unique identities allocated to them, and the derived secret keys K_i. Please note that the keys K_i derived from K_C are computationally independent since the used PRF function has strong one-way property [6]. Thus, even if one device S_i (except the coordinator) is captured, the master secret key K_C as well as other derived keys are safe; in Section 3 we provide a more detailed security analysis. The light source unit (LS) completes its task at this stage of LIRA protocol and it can safely delete/forget all the session keys and identities. The key generation process is shown in Fig. 13 given in Appendix.

2.2. Key verification over a radio channel

To verify the correct and authentic key reception in the previous stage, the network devices use a public radio channel. We use the second radio channel because the light channel used in LIRA only allows one-way communication (from LS to $S_i, i \in \mathcal{G}$). For this purpose, in LIRA we adapt ISO/IEC 9798-4-4 three-pass mutual authentication protocol. The goal of this part of LIRA is to allow the coordinator C to mutually authenticate with other group devices $S_i, i \in \mathcal{G}\backslash\{C\}$, using the keys received

[§] Being a web application, the user would simply have to load a given web page, using `https` protocol, on his/her touchscreen device.

over the visible light channel. Please note that LIRA does not provide for mutual authentication between the devices $S_i, i \in \mathcal{G}\backslash\{C\}$; they can easily accomplish this later, by using the coordinator C as the common trusted party.

The key verification goes as follows. Referring to Fig. 2 (the solid arrows), the coordinator C generates a random n-bit nonce N_i and sends to the device S_i the message flow $\{ID_i, N_i\}$. Upon the reception of the message, S_i generates a random n-bit nonce N_{Si}, and sends back to C the following message flow $\{N_{Si}, [N_{Si}, N_i, \mathtt{ID_C}]_{K_i}\}$, where $[\mathtt{msg}]_K$ denotes that the integrity and authenticity of $\mathtt{msg}$ is ensured through the use of a PRF keyed with K (i.e., a MAC function). On receipt of this message flow, C verifies the authenticity of the message using the key K_i that C can derive from the device identifier ID_i and the master key K_C using an appropriate PRF. If C correctly verifies the received message, it knows that S_i holds the correct key K_i. Finally, C closes the protocol by sending the message $[N_i, N_{Si}]_{K_i}$ back to S_i that in turn verifies the message authenticity using the key K_i and the knowledge of N_{Si}. A successful verification of this last message is indicated by a green LED powered ON on the device S_i.

The coordinator C repeats the above protocol with each device $S_i, i \in \mathcal{G}\backslash\{C\}$. After the correct key verification the green LED will be powered ON on all the devices from the group $\mathcal{G}$ (including the coordinator); for more details please refer to subsection 2.4. To successfully conclude the bootstrapping process, the user has to press a given button on all the devices, within a predefined time period (e.g., 10-20s), but only after having verified visually that LEDs on all the devices (including the coordinator) are ON; this is done in *all-or-none* fashion. In Section 3 we give a detailed security analysis of LIRA protocol.

2.3. Simplified LIRA protocol (LIRA+)

Here we show how to accomplish the same authentication goals as with the basic LIRA protocol, but with a smaller number of messages. We term this simplified version of LIRA protocol as LIRA+. LIRA+ is given in Fig. 3; the part of the protocol relating to the key generation and transmission over the visible light channel is the same as in the basic LIRA and hence not shown here. Being a two-pass mutual authentication protocol, LIRA+ reduces overall number of messages by $(G - 1)$ compared to the basic LIRA, G being the group size. Moreover, LIRA+ does not employ random nonces, thus reducing the computational cost on each network device (in particular on the coordinator). This is made possible by the following two facts: (i) the master key K_C, as well as the keys K_i ($i \in \mathcal{G}\backslash\{C\}$) derived from it, are all freshly and randomly generated at the beginning of a bootstrapping session, and (ii) the device identities are unique in each session. In Section 3, we formally establish the security guarantees of LIRA+ protocol using Scyther tool[¶].

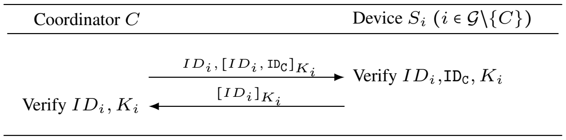

Coordinator C
Device S_i $(i \in \mathcal{G}\backslash\{C\})$
$ID_i, [ID_i, \mathtt{ID_C}]_{K_i}$
Verify $ID_i, \mathtt{ID_C}, K_i$
$[ID_i]_{K_i}$
Verify ID_i, K_i

Figure 3. LIRA+ protocol.

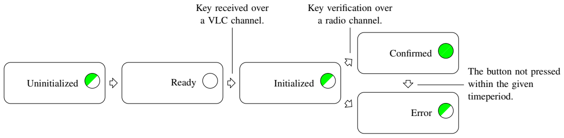


Figure 4. Possible device states and transitions between them; a colored circle indicates that the device LED is ON, while a half colored that the LED is blinking.

2.4. Indicating a device status to the user

As the user needs to know if a bootstrapping process was successful for each network device, the current state of a device will be indicated with a LED according to the state diagram shown in Fig. 4. There are four possible states that a device can occupy: *Uninitialized*, *Ready*, *Initialized* and *Confirmed or Error*.

Uninitialized state. When a device is first powered on it occupies the uninitialized state, which is indicated by the LED blinking once.

Ready state. After the initial blink the device will enter the ready state with the LED turned OFF, and it is ready to receive a session key through the light channel. The device will remain in this state indefinitely and will only proceed after it has received an appropriate synchronization/start message over the light channel. Following this message the device expects to receive a secret session key along with other parameters defined by LIRA protocol, over the same channel.

[¶] http://www.cs.ox.ac.uk/people/cas.cremers/scyther/

DOI: 10.1002/sec

Initialized state. Having received a secret key K_i over the light channel, the device enters the initialized state. In this state the green LED will blink continuously to indicate that the key verification process over a public radio channel is in progress.

Confirmed or Error state. If the key verification was successful, the device enters the confirmed state, indicated by the LED powered ON, awaiting the user to confirm the bootstrapping process by a push on the device button. If the button is not pressed within a predefined time period (e.g., 10-20s) or the key verification phase was not successful, the device enters the error state indicated by the device LED blinking continuously with two times higher frequency than in the initialized state. In this case, the the device automatically resets itself and the user has to repeat the bootstrapping process with all the network devices; if needed the user can reset any device by a long push on the device button.

3. SECURITY ANALYSIS OF LIRA

In this section we provide a detailed security analysis of both LIRA and LIRA+ protocols, with respect to different attacker capabilities.

Security of a visible light communication (VLC) channel. Security of both LIRA and LIRA+ protocols relies heavily on the extent to which we can block the access to the session keys K_C and K_i, $i \in \mathcal{G} \backslash \{C\}$, transmitted over the VLC channel. For this reason, we require a network device to physically occlude the flashing part of the screen below it and to completely block any light coming from this part. In this way, we can protect the session keys not only from eavesdropping, but also from possible modifications when sent over the VLC channel. To make the eavesdropping task more challenging, at least for a camera-equipped adversary, the screen could emit a random interfering light pattern around the position occupied by the device, simultaneously with the regular light signal. A more powerful adversary, capable of analyzing potential electro-magnetic emanations from electronic circuits and video cables [23], can easily break both LIRA and LIRA+. For this reason, in Section 4 we propose a bootstrapping protocol that is secure in this powerful adversary model.

Radio channel. We consider Dolev-Yao adversary model in which an adversary has a full control over a radio channel (he is capable of eavesdropping, modifying, blocking any message transmitted over a radio channel). To verify whether certain secrecy and authenticity properties are satisfied in LIRA and LIRA+ protocols in this model, we use Scyther tool [30]. Scyther is a tool for the symbolic automatic analysis of security properties of security protocols (i.e., confidentiality or different variants of authenticity). It assumes perfect cryptography, meaning that an attacker learns no private information from encrypted messages unless he knows the encryption key. Using Scyther tool we performed bounded verification of LIRA and LIRA+ protocols under Dolev-Yao attacker model, where the *bound* refers to the number of parallel protocol runs executed by honest entities (the network devices in our case). In our analysis we used five honest entity runs; according to the author of Scyther [30] *attacks such as replay or man-in-the-middle attacks are typically found within the bound of two or three runs for many protocol*. Our choice to use Scyther tool for security analysis was motivated by the fact that LIRA protocol is based on adopted ISO/IEC 9798-4 three-pass mutual authentication protocol that was also analyzed with Scyther by Basin et al. [31].

In our security analysis we are interested whether the following authentication and secrecy properties are satisfied: *secrecy* of the session keys K_C and K_i, $i \in \mathcal{G} \backslash \{C\}$, and *injective agreement*. In [32] Lowe provided a hierarchy of authentication specifications, with *injective agreement* being the strongest one.

Definition 3.1 (Injective Agreement [32])
We say that a protocol guarantees to an initiator A agreement with a responder B on a set of data items *ds* if, whenever A (acting as initiator) completes a protocol run, apparently with responder B, then B has previously been running the protocol, apparently with A, and B was acting as a responder in his run, and the two agents agreed on the data values corresponding to all the variables in *ds*, and each such run of A corresponds to a unique run of B.

The implementations of LIRA and LIRA+ protocols in Scyther are given in Appendix. Please note that any *send* event in Scyther is dispatched over the Dolev-Yao channel; thus an attacker can learn any non-encrypted message within send events. For this reason, messages transmitted over the secure VLC channel (i.e., session keys in LIRA and LIRA+) are not modeled as send events because they are not subject to compromise in our model. In Scyther implementation of LIRA and LIRA+, we model computationally independent session keys K_i, $i \in \mathcal{G}\backslash\{C\}$, as random long term secrets shared between the coordinator C and each network device. This is justified by the fact that the session keys are derived using a PRF applied to the master key K_C and unique device identities. Furthermore, we focus our analysis on the two-party settings, involving only the coordinator and one network device. Indeed, in LIRA and LIRA+ protocols, in the given bootstrapping session, the coordinator essentially runs a two-party authentication protocol with every other device in the network, using computationally independent session keys between different protocol runs. If an attacker could exploit the knowledge of a session key K_i of one honest device against another honest device with a computationally independent session key K_j (in the same session), this advantage would transform directly into the advantage against the employed PRF.

Verification of LIRA. Scyther validates that the model of LIRA protocol (given in Appendix) that involve less than six honest device runs satisfies *injective agreement* and preserves *secrecy* of the session key K_i. Please note that in Scyther, the injective agreement property is satisfied only if both *non-injective agreement* and *non-injective synchronization* hold [32]. Furthermore, the bound on the number of honest device runs is not a limiting factor in our analysis since all the keys and random nonces in LIRA are assumed to be of sufficient length (e.g., 128 bits), so that a polynomially bounded attacker can gain only a negligible advantage by observing more protocol runs.

Verification of LIRA+. Unlike for LIRA, Scyther validates that the model of LIRA+ protocol (given in Appendix) that involve less than six honest device runs satisfies only *non-injective agreement* and preserves *secrecy* of the session key K_i, but does not satisfy *non-injective synchronization*. Scyther outputs a trace describing an attack on *non-injective synchronization* property. The attack involves interleaving of messages between different runs of LIRA+ protocol on the given pair of honest network devices. Recall however that in LIRA+ all the protocol messages are authenticated with a fresh session key (see Fig. 3). Therefore, for this attack to work the given pair of devices should use the same session key between different protocol runs. But in LIRA+ this precondition is not fulfilled because the session keys are freshly and randomly generated in each protocol run. Therefore, this attack does not really apply to LIRA+, because it is not possible to interleave LIRA+ messages between different protocol runs. Therefore, LIRA+ also satisfies *non-injective synchronization* property and hence the strongest authentication guarantee - *injective agreement*.

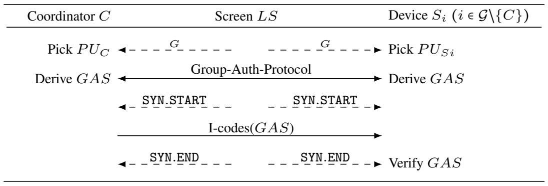


Figure 5. Public-key cryptography based key deployment scheme. Dashed arrows indicate a transmission over the secure visible light channel, while solid line arrows represent the communication over the public radio channel.

Device capture. If the attacker gets in the possession of an initialized network device, he can potentially extract the keying material stored in the device's memory. Against this type of attack there is not a completely secure solution. Depending of the application scenario, one can potentially equip the devices with tamper-resistant units. Also, the scope of damage is different depending on whether the attacker compromises the coordinator device (holding all the session keys) or one of the other honest devices (holding just its own session key).

4. PUBLIC KEY BASED DEPLOYMENT SOLUTION

In this section we extend the attacker model to a more powerful adversary who can also observe the electromagnetic radiation from the screen and/or from the cable that connects the video card with the screen [23]. By observing these emanations the attacker can easily extract all secret keys K_i deployed to devices.

To counter such adversaries, we explore the possibility of using the family of protocols that enable mutual authentication of public keys exchanged by the devices over an insecure radio channel that result in a public authentication value/string [33], [16], [34]. More specifically, our proposed public key-based solution is based on I-codes, a physical layer primitive for authenticated string comparison over the insecure wireless channel [24]. To better understand the security and robustness of the proposed solution, we also study in detail the behavior of I-codes in realistic (low and high interference) environments both analytically and through extensive experiments.

The main motivation to use I-codes for transmission of public messages rather than using approaches in which users verify public messages communicated via VLC [25], [16] is to design a solution that minimizes participation by non-specialist users.

4.1. Description of the public key deployment scheme

The proposed public key deployment scheme is shown in Fig. 5. As before, a user wants to initialize a group $\mathcal{G}$ of wireless devices ($\mathcal{G} = \{1, 2, \ldots, M\}$). The user designates an arbitrary device as a coordinator C by placing it on the touchscreen so that the touchscreen can learn its position. The user then places all other devices S_i, $i \in \mathcal{G}\backslash\{C\}$, so that the multitouch screen (*LS*) can learn their position, as well as the total number of devices involved in the key deployment. More precisely, upon placing the devices on the screen, the user initiates, with a push on the touchscreen's button, the transmission of the group size $G = |\mathcal{G}|$ via VLC (blinking screen) to each group device.

It is essential for security of any group device pairing protocol from this category of protocols that at least one uncompromised device knows the exact number of devices involved in the initialization [16], [28]. If this condition is not met the attacker could easily add its own device to the group. Although some solutions propose users to enter the group size into every device, the problem cannot be easily addressed, especially in scenarios where devices lack user interfaces (such as screen, keyboards, keypads...). For this reason in our solution the touchscreen transmits information about the group size using VLC to each group device.

Having received information about the group size from the screen, the devices begin with Group Authentication Protocol over a radio channel (Fig. 5). During this phase the coordinator C initiates the execution of an authentication protocol such as GAP [16] or SAS [35, 33] to exchange public keys with the devices S_i. The protocol results in a public authentication value/string, such as a *hash* of public keys [36], a *short authentication string* (SAS) [35, 33] or a *group authentication string* (GAS) [16] (Fig. 5).

To ensure the integrity of the resulting public authentication value/string, in our key deployment scheme we use VLC in combination with a security primitive *integrity codes* (I-codes) [24]. The key characteristics of I-codes are: unidirectional message coding, on-off keying modulation and receiver's awareness of presence in the sender's transmission range. Unidirectional error detecting codes are able to detect any number of unidirectional errors in a given code word; for example, it is possible to change a bit "0" into a bit "1" but the contrary is not possible. In I-codes, on-off keying modulation is achieved such that the bit "1" is transmitted on the channel as the presence of a radio signal and the bit "0" is transmitted as the absence of signal. This modulation scheme along with unidirectional message coding enables authenticated string comparison over insecure radio channels given that the adversary cannot annihilate/cancel the radio signal (bit "1") - so called "anti-blocking property" [24]. In original I-codes all devices are synchronized with respect to the SYN message sent over the radio channel by the coordinator C [24], whereas, in our implementation a touchscreen uses a VLC channel to transmit the SYN message to the devices. The purpose of the SYN message is to make devices aware that the transmission of the I-coded authentication string over a radio channel has started.

As shown in Fig. 5 the beginning of the public authentication string transmission starts with a synchronization message (SYN) sent over VLC. This synchronization signal is sent by the light source *LS* to all the devices. After the coordinator C and the devices S_i read the `SYN.START` message, the coordinator C starts to broadcast the group authentication string (GAS) over the radio channel using the I-codes. The coordinator C first encodes the group authentication string (a binary vector of size ℓ) using Manchester coding ($0 \rightarrow 01$ and $1 \rightarrow 10$) and obtains a vector of size $2 \cdot \ell$ bits. After that, every bit is sent over a radio channel using on-off keying (i.e., bit "1": short packet with random content, bit "0": no signal). At the same time, the devices S_i detect the existence of the message between the two synchronization messages (`SYN.START` and `SYN.END`) via threshold energy detection and verify if the message received over a radio equals the one established during the execution of the group authentication protocol over the radio. If the verification is successful then the device S_i

holds the authenticated public keys from all other group devices. To inform the user that the device initialization process was successful a green LED is powered ON. When the green LED powers ON on all group devices the initialization process is finished.

4.2. Security of VLC channel in a stronger attacker model

In this section we extend the attacker model to a more powerful adversary who can inject its own light signals in the VLC channel between the screen and the group device. Here we consider a scenario in which an attacker can inject a light beam almost parallel to the screen (e.g., using a laser) so that the adversary's light is reflected to the device's photodiode, even if the authentic blinking areas on the screen are fully covered by the devices. In this way, the attacker can violate the integrity of messages transmitted over VLC from the screen to the device.

We can easily mitigate such attempts by an adversary by applying I-codes to the VLC channel. Please note, to convey information via VLC we use on-off keying (i.e., bit "0": screen OFF, bit "1": screen ON). With such keying the attacker can modify messages by flipping $0 \rightarrow 1$, but not vice versa ($1 \rightarrow 0$), as the attacker cannot force a powered ON screen to power OFF. In this case we speak of *semi-authenticated* visible light channel. Following I-codes, we first Manchester encode a message to be transmitted over the VLC channel ($0 \rightarrow 01$, $1 \rightarrow 10$). As a result, the transmitted sequence contains an equal number of 0s and 1s. Since the attacker can only flip 0 to 1 (power the light ON), any manipulation of messages over VLC by the attacker would result in an unequal number of 0s and 1s. To protect the integrity of SYN messages, as well as group size message in our protocol (Fig. 5), the start of the VLC message transmission is indicated by changing the initial light intensity of the flashing area (the area below the device) from high intensity to low intensity. Thus, prior to the start of transmission over VLC (SYN messages or group count message) the initial screen light intensity (part of the screen below the device) will be high. With such message encoding, the attacker cannot manipulate the start of the SYN message, nor can modify the group size message. However, the adversary can still convert symbols $0 \rightarrow 1$, in which case, the receiver will simply drop the received message. For example, let us assume that the original `SYN.START` codeword is 010110 and that the adversary flips the first symbol $0 \rightarrow 1$. This will result in the codeword 110110. However, such a codeword cannot be demodulated, because there is no transformation for the pair 11; it will simply be ignored by the receiver.

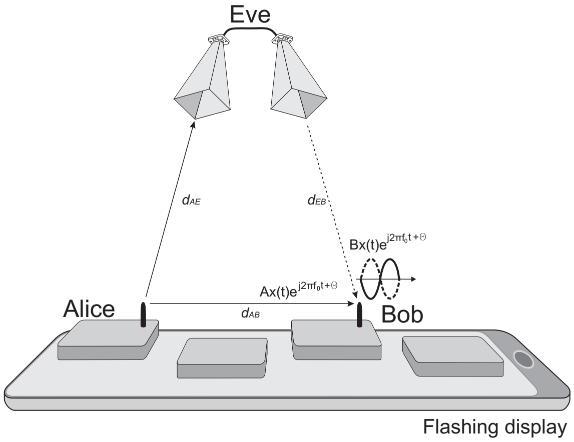
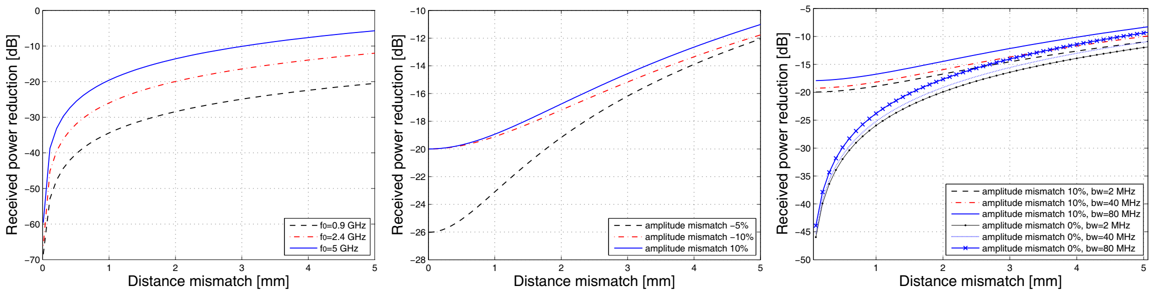


Figure 6. Signal cancellation attack. An example that shows how the attacker (Eve) can cancel the signal sent from one device (Alice) to another (Bob).

4.3. Signal cancellation attack in realistic environments

In this section we explore the possibility of the the attacker to cancel out the public value/string coded with *security primitive* I-codes by flipping at least one symbol "1" of the string into symbol "0". Due to the on-off keying modulation of I-codes this would imply that the adversary has to annihilate at least one signal (waveform) emitted on the channel [24]. In this section we perform a detailed analysis of the attacker's capabilities to cancel out the original I-codes signal using antenna cancellation model presented by Choi et. al. [37]. Since we study signal cancellation attacks in realistic environments, we give a more detailed analysis of I-codes in realistic environment, both analytically and through experiments.

This form of attack is shown by Pöpper et. al. [38] and it is based on signal annihilation when the attacker successfully sends an inverse signal to the receiver. The attacker does not generate its own signal, but instead utilizes a pair of directional antennas to annihilate the sender's signal at the receiver end, as shown by an example in Fig. 6. Inspired by this work Hou et. al. [27] proposed the solution with frequency hopping to achieve arbitrarily small probability of signal cancellation.



(a) (b) (c)

Figure 7. (a) The received power reduction as a function of the the distance mismatch if the amplitude mismatch is $\Delta A = 0$ for different carrier frequencies. (b) The received power reduction as a function of the distance mismatch for different amplitude mismatches (-5%, -10% and 10%) and the carrier frequency of 2.4 GHz. (c) The received power reduction as a function of the distance mismatch for signals with different bandwidth.

In this paper we show that in realistic environment conditions it is quite difficult for the the attacker to annihilate the signal. More specifically, we assume that the attacker does not know the exact distance between the antennas of two devices (d_{AB} in Fig. 6), nor the exact amplitude of the baseband signal at the receiver's end. More specifically, we show that even a small distance mismatch (e.g., 3mm), and/or small amplitude mismatch (e.g., 10 %) the signal cancellation will fail. Since wireless transmission uses a band of frequencies, we also study the impact of signal bandwidth on signal cancellation.

Fig. 6 shows how different strategies by the attacker (Eve) affect the power of the modulated (cosine) signal sent from one device (Alice) to another (Bob). The attacker will successfully cancel out the signal when the signal amplitudes from Alice and Eve are equal with a phase shift of $\theta = (2k-1)\pi$, $(k = 1, 2, 3, \ldots)$ at the receiver (Bob). This means that the attacker must know the exact location (position) of wireless devices Alice and Bob, as well as the channel conditions. We next show how demanding is for the attacker to meet these conditions.

Let us denote with Δd the distance difference between relay (attacker) and a direct channel between Alice and Bob, i.e. $\Delta d = d_{AE} + d_{EB} - d_{AB}$ (Fig. 6). The phase shift θ between two signals traveling over the two channels satisfies $\theta = 2\pi f_0 \Delta d/c$, where f_0 presents a carrier frequency and c denotes a speed of light. The optimum position for the attacker in order to cancel the signal satisfies the following condition $d_k = d_{AB} + \frac{c}{2f_0}(2k-1), (k = 1, 2, 3, \ldots)$. Any difference between the actual position and the closest optimal position of the attacker presents the *distance mismatch* $\delta = |d_a - d_k|$.

We will now study how the mismatch in the amplitude and the phase affect the attacker's ability to cancel the radio signal at the receiver. As shown in Fig. 6 Alice modulates the signal baseband from the $x(t)$ and sends it to Bob, $s_a(t) = A_0 x(t) e^{j2\pi f_0 t}$. This signal is attenuated and phase shifted in the wireless channel and hence received by Bob as $s_b(t) = Ax(t)e^{j2\pi f_0 t}$, where A and f_0 are the amplitude and center frequency of the received original signal, respectively. Due to simplicity of calculation and without any loss of generality, we assume that there is no phase shift between signals s_a and s_b. The attacker Eve, using two directional antennas, relays the signal $s_a(t)$ to Bob. This signal is received by Bob as $a_b(t) = Bx(t)e^{j2\pi f_0 t + \theta}$, where B, θ, and f_0 are the amplitude, the phase shift and the center frequency of the signal relayed by the attacker, respectively. Bob receives the following composite signal:

$$r(t) = s_b(t) + a_b(t) = Ax(t)e^{j2\pi f_0 t} + Bx(t)e^{j(2\pi f_0 t + \theta)}. \tag{1}$$

Let us express the attacker's signal amplitude as $B = A + \Delta A$, where ΔA represents the amplitude mismatch between the original and the attacker's signals at Bob. Please note that here we ignore the noise effect thus making the job easier for the attacker. The instantaneous power of the received signal is [39]:

$$\begin{aligned} P_r(t) &= r(t) \cdot \overline{r(t)} \\ &= 2A(A + \Delta A)|x(t)|^2(1 + \cos(\theta)) + (\Delta A)^2|x(t)|^2, \end{aligned} \tag{2}$$

where $\overline{r(t)}$ is the complex conjugate of the signal $r(t)$ received by Bob, $|x(t)|$ is the absolute value of the baseband signal $x(t)$.

From Eq. 2 we can see that the attacker will successfully cancel out the original signal $s_b(t)$ if the amplitude mismatch between the direct signal and the relay signal is ΔA = 0, and the phase shift θ is multiple of π.

Fig. 7(a) shows the received power reduction as a function of the distance mismatch δ for carrier frequencies of 0.9 GHz, 2.4 GHz and 5 GHz. For the optimal position of the attacker (δ = 0 mm) the received power reduction are 68.4, 65.97 and 60 dB for the carrier frequencies 0.9, 2.4 and 5 GHz frequencies, respectively. Please note, if the distance mismatch is only 3 mm the power reductions will be 25, 16.5 and 10 dB for these frequencies.

As can be seen in Fig. 7(b) the received power reduction will be even smaller if we consider both mismatches in the amplitude and phase of the attacker's signal. For example, if the distance mismatch is 3 mm the received power reduction will be 16.1, 15.1 and 14.56 dB for the amplitude mismatches of only -5%, -10$\%$ and 10$\%$ and the carrier frequency f_0 of 2.4 GHz.

All these considerations have been carried out for the optimal position of the attacker according to the center frequency, but there is a small mismatch in the optimal position for other frequencies within the signal bandwidth. To observe the effect of signal bandwidth we generated QPSK modulated signals and calculated the received power for signals with different bandwidths using Matlab. As can be seen in Fig. 7(c), the power reduction at the receiver will be smaller for the signal with wide bandwidths. For example, if there is no the distance mismatch (δ = 0 mm) and with perfect amplitude matching (i.e. ΔA = 0) then we have the power reduction of -46 dB, -45.5 dB and -44 dB using 2 MHz, 40 MHz and 80 MHz bandwidths. If the amplitude mismatch is 10$\%$ and there is no the distance mismatch (ΔA = $0.1A$ and δ = 0 mm) then we have the power reduction of 19.9 dB, 19.25 dB and 17.92 dB using 2 MHz, 40 MHz and 80 MHz bandwidths. Now if we suppose the attacker's position off 3 mm from the optimal position (δ = 3 mm) then the power reduction becomes 14.5 dB, 13.7 dB and 12.1 dB using these bandwidths.

Please note that in our initialization scheme radio transceivers emit signals with 0 dBm transmit power around the carrier frequency of 2.4 GHz, and the power of the received signal is between -30 dBm and -40 dBm (the group devices are located in close vicinity of each other). The standard receiver sensitivity for this frequency band is about -95 dBm. The results of the performed analysis indicate that in the best scenario the attacker can attenuate the signal at the receiver up to 19.9 dB for 0 mm distance mismatch and 10$\%$ amplitude mismatch using signal with bandwidth of 2 MHz. In other words, even in such very advantageous scenarios for the adversary he will not be able to sufficiently attenuate the radio signal at the receiver's end. This, and the fact that it is very likely the attacker will not know the exact distance between himself and the group devices, nor the originating signal's power at the receiver end, we can conclude that in realistic conditions the probability to cancel the signal at the receiver is negligible. We can note that all these considerations were taken under very advantageous assumptions for the attacker: no multipath fading effects, no noise in the radio channel, complete knowledge of directivity and gains of antennas. Also the attacker must be located close to the direct channel and act very quickly in order to cancel the signal by using two coupled directional antennas. Longer distances would require additional amplification of the signal relayed by the attacker, implying the usage of active electronic devices which affects the signal spectrum and also complicates the signal cancellation. In conclusion, under the assumption of our system model, the energy of the radio signal cannot be annihilated by an adversary at the receiver and thus radio signal anti-blocking property can be safely assumed in our solution.

4.4. Performance of public key deployment in high interference environment

An important practical consideration that has not been explored yet is the performance of I-codes in conditions of non-malicious interference from other wireless devices operating within our frequency spectrum. Capkun et al. [24] performed a robustness analysis of I-codes which shows the message transmission success ratio as a function of the size of transmitted messages. Hou et al. [27] considers non-malicious interference from other nearby wireless devices operating within the same frequency spectrum, and proposes a solution based on uncoordinated frequency hopping. Similarly to Hou et al., in this paper we also study jamming impact of non-malicious interference on our signal under conditions of low and high interference. However, we show how combination of radio channel with an out-of-band visible light channel for synchronization can make our solution robust in high interference conditions, even without applying frequency hopping mechanisms.

Implementation of I-codes. We implemented I-codes on Arduino Uno microcontroller board based on the ATmega328 and nRF24L01+ single chip 2.4 GHz radio transceiver. This transceiver is suitable for ultra low power wireless applications and operates in frequency band at 2.400 - 2.4835 GHz using GFSK modulation. It has a programmable output power (-18 to 0 dBm), and a receiver sensitivity of -94, -85 and -82 dBm at 250 kbps, 1 and 2 Mbps, respectively. In our implementation of I-codes we did not use CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) mechanism to get more realistic results under conditions of high interference.

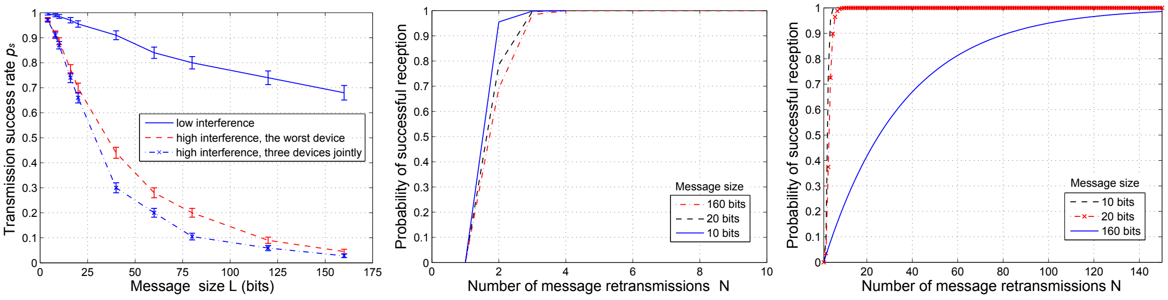
In our implementation of I-codes original message m is first Manchester coded ($0 \rightarrow 01$ and $1 \rightarrow 10$), and then transmitted over a radio channel. Binary “1” is transmitted as a packet containing a random payload of 32 bytes. The packet is transmitted in 0.4 ms at 1 Mbps. On the other hand, binary “0” is transmitted as an absence of signal of duration 1.6 ms. Thus, the overall time required to transmit a singe bit of the original message is 2 ms. At the receiver’s side, if the signal power is below a pre-set RSSI level of -64 dBm during the period of 1.6 ms it is decoded as bit “0”, whereas the presence of signal is decoded as bit “1”.

Results of experimental analysis. In the first scenario we studied the impact of low interference from other signals on our I-codes implementation. Besides our radio signal at 2.448 GHz (WiFi channel 7) there were signals from 3 WiFi networks that operated on channels 6, 9, and 11. In our experiments we used 2 nRF24L01+ devices that were placed on the screen. Recall, the screen is used to synchronize the transmission of I-codes encoded messages using VLC channel. One device acted as a source of I-coded messages and the other one as the destination. The source generated and transmitted messages of varying size (from 4 bits to 160 bits). The source repeatedly transmitted each message 1000 times. Fig. 8(a) shows the message transmission success rate p_s (obtained as the ratio between the number of correctly received messages and the number of total transmitted messages).

After that we performed the experiments in a high-interference scenario. To accomplish high interference environment, in addition to our radio signal at 2.448 GHz, there were signals from 6 Wi-Fi networks that operated on channels 5, 6, 9 (two networks) and 11. We placed in the vicinity of our devices (about 3 meters away) an access point that operated on channel 7 (2.442 GHz) and continuously transmitted packets. To increase the effect of interference we also exchanged a large amount of data using the Bluetooth communication between the two Bluetooth devices placed 2 meters from our devices. In this example we used a group of 4 devices that were placed on the screen to synchronize the transmission process by visible light (Section 4). One device transmitted messages of varying size encoded using I-codes, each message was repeated 2000 times. In this case we measured the message transmission success ratio p_s for each of the three remaining devices. In Fig. 8(a) we show the results for the device that is most affected by interference, as well as the total success ratio for all three receiving devices.

In Fig. 8(a) we can see that the transmission success rate p_s decreases rapidly as the message size increases, especially in the case of high interference, meaning that I-codes are best suited for reasonably short messages. If we want to transmit a longer message under the same conditions in the channel it would be necessary to retransmit the message a larger number of times. This would result in a significant increase in the total message transmission time. To mitigate this problem we suggest breaking the message in smaller fragments of fixed size. We show that in this way we can make our solution very resistant to non-malicious high interference.

Message fragmentation. Before the transmission, the coordinator device C splits the original message m into n fragments ($m = m_1, m_2, \ldots, m_n$) each size of $\ell = \lceil \frac{|m|}{n} \rceil$ bits. After that the coordinator sends every fragment m_i repeatedly over the radio in the sequence $m_1, \ldots, m_n$. As before, fragment transmission and reception between the coordinator C and the remaining group devices $S_i,\ i \in \mathcal{G} \backslash \{C\}$ is accomplished through the VLC channel. The touchscreen synchronizes the fragment transmissions by sending the fragment number j over the VLC ($j \in \{1, \ldots, n\}$) to each group device. The transmission of message fragments over the radio and the VLC channel alternates as follows:



(a) (b) (c)

Figure 8. (a) Transmission success ratio p_s as a function of the transmitted message size for the case of low and high interference. (b) Probability that a message is successfully received under conditions of low interference. (c) Probability that a message is successfully received under conditions of high interference.

$$\underbrace{\boxed{1}\boxed{m_1}\cdots\boxed{n}\boxed{m_n}}_{1^{st}\text{ transmission}}\cdots\underbrace{\boxed{1}\boxed{m_1}\cdots\boxed{n}\boxed{m_n}}_{2^{nd}\text{ transmission}}\cdots$$

Here the colored squares represent the transmission over the VLC, which are followed by the transmission over the radio denoted with white squares. On the reception side the receiver device S_i assembles the fragments according to the fragment number. As we show in the sequel, given a sufficient number N of retransmissions, each device S_i will eventually receive all n fragments.

Estimating the minimum number of fragment retransmissions. Let us denote with Y the number of times the coordinator has to retransmit a message m such that the message is successfully received by all devices. The probability that a message is successfully received after N retransmissions can be calculated as $P[Y \le N] = (1-(1-p_s)^N)^n$, where p_s is a message transmission success ratio (obtained from our experiments) and n is the number of fragments. In Fig. 8(b) and Fig. 8(c) we show the probabilities that 10, 20 and 160 bit long messages (I-coded messages of 20, 40 and 320 bits) are successfully received after N retransmissions, under conditions of low and high interference.

From Fig. 8(b) we can see that under conditions of low interference the best way would be to transfer the entire message without dividing it into fragments. For example, if we want to transmit a 160 bit message (a typical size of the message digest) in a single transmission there would be 70% chance of message being transmitted correctly under these conditions. Likewise, for the targeted success probability of $99,9\%$ we require 3 successive transmissions, as shown in Fig. 8(b). Recall from Section 4.4, in our implementation it takes a total time of 2 ms to transmit one bit of the original message. This means that overall 960 ms (160 bits×2 ms×3 retransmissions) is required to successfully transmit a 160 bit long message. However, under conditions of high interference there is a small chance to successfully transfer a 160 bit message without dividing it into fragments, as can be seen in Fig. 8(c).

Our goal is to find an optimum (minimum) number of fragments n for successful transmission of the complete message m with high probability within the shortest period of time (i.e., the smallest number of repetitions N). The probability that a message is still incomplete after N (re)transmissions is:

$$P[Y > N] = 1 - P[Y \le N] = 1 - (1-(1-p_s)^N)^n. \tag{3}$$

Please note here that we assume all fragment success/loss events to be mutually independent. Then we can calculate the expected total number of (re)transmissions to successfully transfer the complete message:

$$\begin{aligned} E[Y] &= \sum_{N=0}^{\infty} P[Y = N] \cdot N \\ &= \sum_{N=0}^{\infty} P[Y > N] \\ &= \sum_{N=0}^{\infty} (1 - P[Y \le N]) \\ &= \sum_{N=0}^{\infty} (1-(1-(1-p_s)^N)^n). \end{aligned} \tag{4}$$

DOI: 10.1002/sec
Prepared using secauth.cls

After applying the derivative for geometric series we obtain:

$$\begin{aligned}
E[Y] &= \sum_{N=0}^{\infty}(1-\sum_{i=0}^{n}\binom{n}{i}(-1)^i(1-p_s)^{Ni}) \\
&= -\sum_{N=0}^{\infty}\sum_{i=1}^{n}\binom{n}{i}(-1)^i(1-p_s)^{Ni} \\
&= -\sum_{i=1}^{n}(-1)^i\binom{n}{i}\sum_{N=0}^{\infty}(1-p_s)^{Ni} \qquad (5)\\
&= -\sum_{i=1}^{n}(-1)^i\binom{n}{i}\frac{1}{1-(1-p_s)^i} \\
&= \sum_{i=1}^{n}\binom{n}{i}\frac{(-1)^{i+1}}{1-(1-p_s)^i}.
\end{aligned}$$

The total time required for the transmission of a single message equals $T = T_{VLC} + T_r$, where T_{VLC} is time required for the transmission of synchronization signals and the fragment numbers over the visible light channel, and T_r denotes the time required to transmit the message m over the radio channel. In our implementation a fragment number is represented with 8 bits. Since the fragment number is transmited over the VLC channel and one bit over it takes approximately 16 ms we have $T_{VLC} = n \cdot 8 \cdot 16$ [ms]. On the other hand, time T_r depends on the number of fragments n and their size in bits ℓ. As stated earlier in our implementation $2ms$ is required to transmit a single bit of the original message over the radio channel which implies $T_r = 2 \cdot n \cdot \ell$ [ms]. It follows: $T = n \cdot (128 + 2 \cdot \ell)$.

Fig. 9 shows the expected time $E[T] = T \cdot E[Y]$ required to successfully transmit the message as a function of the fragment size ℓ under conditions of low and high interference. From Fig. 9 we can see that under conditions of low interference the optimal solution is to transfer the entire message without fragmentation (i.e., $n = 1$). The fragments from 10 to 20 bits are best suited for transmission under conditions of high interference. For very short fragments ($\ell < 10$ bits) the minimum expected time required to transmit the message is larger than 6 seconds ($E[T] > 6$ s). This occurs due to the large communication overhead over the VLC channel. On the other hand, for fragments larger than 20 bits ($\ell > 20$ bits) the transmission success ratio p_s significantly decreases, resulting in a significant increase in the expected time required to successfully transmit a message. From Fig. 9 we also see that the optimal fragment size is 20 bits and minimum expected time $E[T]$ required to successfully transmit a message is 4.05 seconds. Thus the original message of size 160 bits will be successfully transmitted within 10.75 seconds in fragments of 20 bits with the success probability of 99.9% under the conditions of high interference ($n = 8$ fragments, $\ell = 20$ bits, $N = 8$ retransmissions).

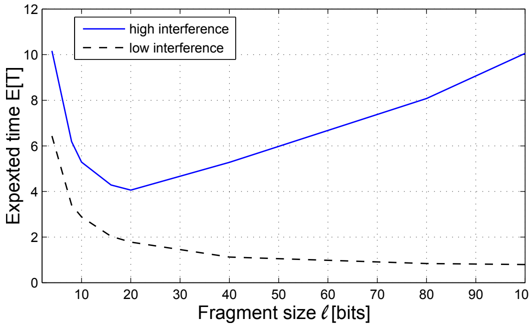


Figure 9. The expected time in seconds required to successfully transmit a message as a function of the fragment size in bits under conditions of low and high interference.

4.5. Discussion

One significant implication of the synchronization based on the visible light channel as well as the transmission of the fragment number over the radio channel is that receivers do not have to know in advance the length of the message to be transmitted over the radio channel. Moreover, if one of the receivers does not correctly receive kth message fragment in ith transmission, it is only sufficient to be correctly received in jth transmission ($j > i$). Consideration conducted in this paper show that our implementation of I-codes is suitable for the transmission of short authentication strings even under the conditions of high (non-malicious) interference. It is also shown that our solution is suitable for the transmission of long authentication messages (e.g. 160 bit message digests) over the radio channel under these conditions.

5. USABILITY EVALUATION

We carried out experiments to study different usability aspects of our solution. We wanted to test the hypothesis that our solution is practical, easy to use, and user-friendly for an end user. We also wanted test the hypothesis that our solution is faster than the initialization of the wireless devices via a cable, and that it scales well with the number of devices to be initialized. Although our usability evaluation is conducted for the symmetric-key cryptography-based protocol (LIRA), the results and conclusions obtained from these tests are applicable to our public-key based solution too. This is true because: (i) from the user point of view nothing changes - the user takes the devices, places them on the screen and pushes a button to start the deployment, (ii) the time required for initialization of the network is almost the same for both protocols.

5.1. Testbed implementation details

We next describe a testbed used in the usability evaluation of our secret-key based protocol (LIRA). The coordinator C and the devices S_i in our solution were implemented on commercially available Arduino Uno platform based on: ATmega328, 32 KB flash memory, 2 KB SRAM, 1 KB EEPROM and 16 MHz clock speed. We also used ZigBee radio (Xbee S2 module), BPW34 photodiodes and green LEDs. In our experiments we emulated a multi-touchscreen using regular 22-inch LCD screen placed horizontally. The exact position of the devices S_i as well as their number is detected using a camera located above the screen and connected to the PC running an OpenCV (Fig. 10); OpenCV recognizes device S_i by detecting yellow markers attached to them.

We developed an application for the PC that controls the “touchscreen”. The application is responsible for keys generation and their broadcast in the form of the light to the devices. For transmissions over the VLC channel the bit rate is limited by the screen frame rate (16.66 milliseconds per frame for a standard 60 Hz LCD screen). Prior to the transmission each bit was Manchester encoded and after that transmitted at 20 bps using on-off keying (0 bit - ON, 1 bit - OFF).

At the reception side, device (a photodiode) samples light levels with a every 10 milliseconds. Since the transmission of every bit lasts for 50 milliseconds this means that device will take 5 samples for each bit. We set a threshold level such that the signal/trace received above that level is converted into a binary “1” or, alternatively into a “0”. Next, a convolution function is applied over the received/sampled series of “zeros” and “ones” using the delimiter with mask $\{0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0\}$. As a result of convolving signal with this mask, we obtain a data array with values ranging from 0 to 5, where elements with extremes 0 and 5 are decoded as bits 0 and 1, respectively. Finally, we apply Manchester decoding to recover the original bit-stream. Due to the Manchester encoding our solution is highly robust to desynchronization effects and frame losses during transmission via the visible light channel.

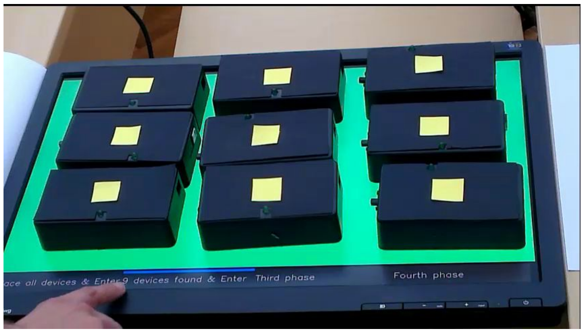


Figure 10. The experimental setup used in our user performance study

Please note that the time required to deploy keys from PC to devices over a light channel is fixed and independent on the number of nodes since the screen simultaneously (in parallel) deploys the keys to each device.

5.2. Test cases and test procedure

We conducted two usability studies: LIRA only and LIRA vs. Cable study. In the first usability study we tested the hypothesis that users perceive the proposed LIRA solution user-friendly and achieve small error rates during the initialization process. The test took about 20 minutes per user. In LIRA vs. Cable study, we tested the hypothesis that users find LIRA solution more user-friendly than the cable-based solution. The test took about 30 minutes per user.

A total number of 48 users took part in the usability study: 22 in the LIRA study and 26 in LIRA vs. Cable study. All participants were volunteers who were asked to help in our research, and none of them was economically motivated to achieve results that are biased towards the positive results of our usability study. They were mainly university students (computer science or electronics), 5 of them had PhD and 2 MSc. For each participant in the study, we logged the overall protocol completion time, as well as the error rate. The completion time included the device manipulative cost (e.g., powering ON the device, placing it on the touch-screen) as well as the total time

Table I. The testers' demographic info as well as their smartphone/tablet device usage.

Age			Gender		Familiar with VLC		Familiar with WSN		Concerned about WiFi security	
<25	25-30	>30	M	F	Yes	No	Yes	No	Yes	No
30	14	4	33	15	15	33	45	3	27	21
Feel secure while using wireless			**Use TS devices (hours/week)**				**Familiar with TS technology (years)**			
Yes	No	Neutral	0-5	5-20	20-30	>30	0-1	1-2	2-3	>3
12	15	21	14	12	8	14	3	10	11	24

Note:TS - TouchscreenIH - Intelligent Home
SP - Smartphone WSN - Wireless Sensor Network
VLC - Visible Light Communication

for key exchange and authentication. The key deployment procedure for each tester was recorded by a camera and the overall times were subsequently extracted from the video. Although our university does not require the ethical review board to review and approve research work with the human testers, all the testers in our study were informed in advance how the collected data of their study will be recorded and processed after the evaluation.

Before proceeding to each test, all the users were briefly introduced to the concept of device pairing. This was followed by the practical demonstration from the administrator and a short video of the procedure itself. The training procedure lasted for about 5 minutes. After that, the users tested the method. At the end of the usability test, the users completed a post-test questionnaire. The System Usability Scale (SUS) [40] test was used to numerically express the users' satisfaction with the system. The System Usability Scale (SUS) is a ten-item (Likert) scale giving a global view of subjective assessments of usability [40]. In addition, the users were also asked for their opinion about the usability and perceived security of the proposed schemes.

5.3. LIRA study

LIRA study was conducted with a total number of 22 participants. Our goal was to verify that the users perceive the proposed LIRA solution user-friendly and can detect errors during the initialization process. An error was indicated by a continuously blinking LED on the device (Fig. 4). To accomplish this the user's task was to initialize a group of 3, 6 and 9 wireless devices. Each tester performed tests 3 times with each group of wireless devices, leading to a total number of 198 test cases (3 × 22 × 3 = 198). In Appendix (Fig. 12) we show an initialization procedure involving 9 devices.

At the beginning of the test, the user powers on the devices and places them on the touchscreen (Fig. 1). Next, the user starts the initialization process (LIRA protocol) and waits for the end of the process which is signaled by LED lights as described in Section 5.1. At the bottom of the screen the user can also observe the current status of the key deployment. During the tests an error was intentionally inserted in one device in a randomly chosen test scenario for every tested group of devices (3, 6 and 9 wireless devices).

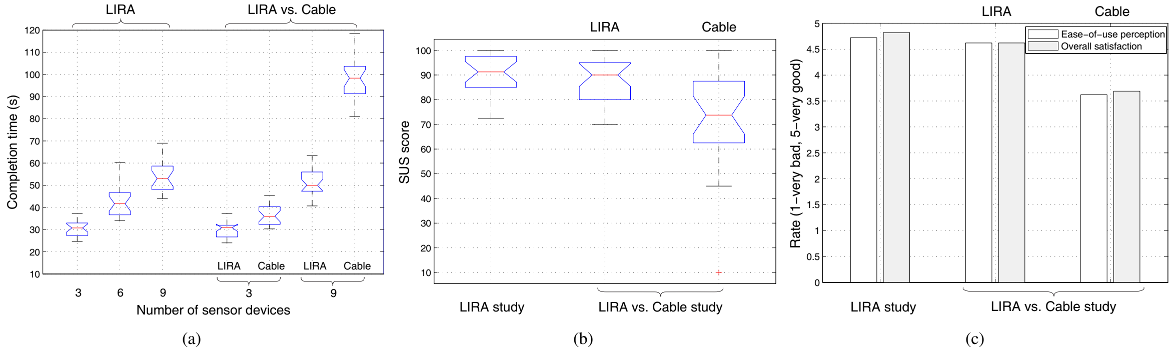
Login time. The total average times required for the successful initialization of 3, 6 and 9 devices were 30.51 (std. 4.33), 42.45 (std. 7.94) and 54.36 (std. 8.71) seconds, respectively, as shown in Fig. 11(a). A higher key deployment (completion) time with a larger group of devices can be explained with the overall time the users are required to power all devices on and place them on the screen; please note from Section 5.1, during the key deployment over VLC, the keys are transmitted in parallel over the screen to every group device.

Error rate. When testing LIRA solution all users were able to identify the error that was indicated by a continuously blinking LED on a randomly selected device.

Usability. Fig. 11(b) shows the SUS score for LIRA solution provided by the users. The average SUS score was 89.77 (out of 100) which makes the solution highly usable [40]. In addition to SUS questionnaire the average user's overall satisfaction rate of the system's features was 4.82 (in a 5-point Likert scale), as can be seen in Fig. 11(c).

5.4. LIRA vs. Cable study

A total number of 26 participants took part in LIRA vs. Cable study. Recall, our goal was to test the hypothesis that users perceive LIRA solution easier to use compared to Cable solution, especially while initializing a larger group of devices. To accomplish this each tester performed tests 3 times with a group of 3 and 9 devices for both solutions (LIRA and Cable) leading to a total number of 312 test cases (3 × 26 × 2 × 2 = 312). The procedure for testing LIRA solution was exactly the same as in the previous study. When testing the Cable solution, the network was initialized by connecting the device to the computer with an USB cable and loading them with the



(a) (b) (c)

Figure 11. (a) Box plots representing protocol completion times accomplished in LIRA study and in LIRA vs Cable study. (b) The resuls of SUS scores for: LIRA solution and LIRA vs Cable study. (c) The average user's ease-of-use perception rate and overall satisfaction for: LIRA study and LIRA vs Cable study.

secret information. Similarly as in the previous study, an error was inserted in one device during the tests to verify the user's ability of detecting errors. As in LIRA study, an error was indicated by a continuously blinking LED.

Login time. The total average times required for the successful initialization of 3 and 9 devices were: 29.87 (std. 4.26) and 51.21 (std. 7.62) seconds for LIRA solution, and 36.68 (std. 7.32) and 98.72 (std. 13.22) seconds for the Cable solution, as shown in Fig. 11(a).

Error rate. When testing LIRA solution all users were able to identify the error. However, in the Cable method the error was not detected in 9 test cases. Five users failed to recognize the error when they tested a group of 9 wireless devices and two users for groups of 3 and 9 wireless devices. After talking to users we learned that the main reason for this was the user's focus during the test. More precisely, while testing the Cable solution the users were more focused on instructions given at the screen (connect the device, key upload status, etc.) but less on the status of the blinking LED on a device that indicated the error, especially while deploying the key to a large group of devices.

Usability. For LIRA solution the SUS score was 87.05, while Cable solution scored 70.68 (out of 100), as shown in Fig. 11(b). In addition we rated the overall user's satisfaction to using LIRA and Cable solutions (in a 5-point Likert scale). The average user's satisfaction rate was 4.62 for LIRA solution and 3.69 for Cable solution, as shown in Fig. 11(c). These results indicate that users prefer LIRA solution over Cable method due to its simplicity and ease-of-use (SUS scores). This result also confirms the results of SUS scores. Furthermore, we can clearly see a correlation between these two parameters: the average rate relating to the user's overall satisfaction and the SUS scores for both solutions. Pearson correlation coefficients between these two parameters is 0.536 and 0.706, and their statistical significance is $p = 8.39 \times 10^{-5}$ and $p = 5.43 \times 10^{-5}$ for LIRA (48 testers) and the Cable (26 testers) solution, respectively.

5.5. Within subject analysis

Paired t-test revealed that the total average times required to successfully initialize 3 ($p = 1.186 \times 10^{-7}$) and 9 ($p = 2.94 \times 10^{-4}$) devices for LIRA solution were significantly smaller than for the Cable solution. Recall, the average task completion times required for the successful initialization of 3 and 9 devices were: 29.87 (std. 4.26) and 51.21 (std. 7.62) seconds in LIRA solution, and 36.68 (std. 7.32) and 98.72 (std. 13.22) seconds for the Cable solution, respectively.

5.6. Between subject analysis

Unpaired t-test revealed that the total average times required to successfully initialize 3 ($p = 1.106 \times 10^{-5}$) and 9 ($p = 2.27 \times 10^{-4}$) devices for LIRA solution were significantly smaller than for Cable solution. Unpaired t-test were applied on a test results with a sample of 22 users for LIRA solution and 26 users for the Cable solution (two different samples). The average task completion times were 30.51 (std. 4.33) and 54.36 (std. 8.71) seconds for a group of 3 and 9 devices, respectively, in LIRA solution. These times were 36.68 (std. 7.32) and 98.72 (std. 13.22) seconds for a group of 3 and 9 devices, respectively, in the Cable solution.

Effect of age. In the performed unpaired t-tests, we did not find any significant effect of age on the task completion time, error recognition, users' ease-of-use perception and their overall satisfaction of our solution. The task completion times for 3 devices using LIRA solution were 30.37 (std. 3.34) and 29.81 (std. 3.18) seconds for age groups of 21-24 and over 24 years, respectively. On the other hand, these times were 52.66 (std. 6.89) and 52.63 (std. 5.24) seconds for 9 devices for the same age groups, respectively. The unpaired t-tests revealed that the total average times required to successfully initialize 3 ($p = 0.288$) and 9 ($p = 0.466$) devices for those age groups were not significantly different. The results of usability evaluation present task completion times conducted with young and educated participants. Note that evaluating secure-usable solutions with such a group of users is a natural first step, and therefore, these results are only a preliminary evaluation. In future work we plan to expand our tests to a more representative and a larger population of users.

Effect of gender. In the performed unpaired t-test, we did not find any significant effect of gender on the task completion, error recognition, users' easy-of-use perception and users overall satisfaction for our solution. The task completion times for 3 devices using LIRA solution were 30.42 (std. 3.42) and 29.57 (std. 2.91) seconds for male and female, respectively. These times were 52.57 (std. 6.07) and 52.82 (std. 6.84) seconds for 9 devices for male and female, respectively. The unpaired t-tests revealed that the total average times required to successfully initialize 3 (p=0.231) and 9 (p=0.417) devices were not significantly different.

5.7. Other observations

According to the results of the post-test questionnaires in the LIRA vs. Cable study 22 users (85%) prefer our LIRA solution over Cable solution. However, a couple of users did not feel comfortable knowing that part of the key deployment protocol occurs over the radio channel. This is mostly because users know that "*...communication can be recorded while traveling over the radio*". Some users also suggested implementing a red blinking LED as an error signalization instead of a green blinking LED to increase the rate of detecting potential errors.

5.8. Initialization of even larger group of devices

Recall, in our tests we used up to 9 prototype-based devices mostly because of their size as well as the screen size. To be more precise, we used a 22-inch size screen and nine 15×7 cm (5.9×2.75 inch) devices. Please note that the number of such devices can be even larger if we take into account that the dimensions of devices may be several times smaller than the ones used in our tests (iBeacons [3]). In addition, our solution allows the initialization of a reasonably larger number of devices (e.g. more than 100). One approach to accomplishing this task would include dividing the devices into smaller groups (e.g. up to 10-20 devices) that can fit the screen size, and initialize every group separately in batches. It is important to note that a single coordinator device (the device that was initially assigned to this role) must be present during the initialization of every group in order to preserve the coexistence of the whole initialization solution. Please note that such approach slightly deteriorates usability since user is now required to repeat the initialization in several batches.

6. RELATED WORK

Many different key deployment schemes such as SPINS [6], LEAP [41], TinySec architecture [42], and schemes [7, 8, 43, 44] have been suggested in the last couple of years. All of these schemes assume that each node already possesses one or more initial secret keys which is not a reasonable assumption for many applications because an attacker can discover the keys during their deployment.

Shake Them Up [45], Smart-Its Friends [46], and Are You with Me [47] are schemes that use movement to establish a secret key and thus require an accelerometer on each sensor node to measure movement. Resurrecting Duckling [48, 49] uses physical contact to set up a secure shared key, however this scheme requires a specialized hardware interface for physical contact. Similarly to the above schemes, Message-In-a-Bottle (MIB) [10] and KALwEN [11] are key management schemes that need additional equipment as a smart faraday cage.

Talking to Strangers [50], Seeing-is-Believing (SiB) [36], MANA I, II, and III [51], Short Authenticated Strings (SAS) [52] and [33] based protocols, MANA IV [53], and schemes proposed by Gehrmann et al. [51], and Wong and

Stajano [54],[12] utilize an OOB channel to setup public keys. The main drawback of these solutions is the number of the exchanged messages over an OoB channel or an insecure wireless channel.

I-codes [35], [24] exploit physical properties of radio channel and consists of three main parts: on-off keying modulation, signal anti-blocking which means that the energy of signal cannot be annihilated by an adversary, and I-coding. Based on I-codes Gollakota et al. in [26] propose Tamper-Evident Pairing (TEP) as a new in-band device pairing protocol for WiFi (IEEE 802.11) devices. TEP uses a tamper-evident announcement (TEA) that protects the message integrity by embedding cryptographic authentication information. Both solutions are one-to-one message authentication primitives suitable for pairwise communication and it is difficult to apply them to securely initialize multiple constrained wireless devices like sensors in WBAN. Inspired by I-codes and TEP Hou et al. in [27] propose Chorus as a scalable in-band trust establishment for multiple constrained devices over the insecure wireless channel. Hou et al. refer to the work of Pöper [38] in which a correlated signal cancellation attack is shown to be practical. However this is a very strong assumption and we are in our work showed through a detailed analysis how it is difficult and almost impossible for an attacker to cancel signal at the receiver in realistic conditions.

Wong and Stajano propose Multichannel Group Device Pairing Protocol (MC-GDP) [34] in which each device has to be capable of demodulating signals received over an OoB channel. In HAPADEP [55] both data and verification information are sent over an audio channel and thus the pairing devices have to be equipped with speakers and microphones. Saxena and Udin in [13], [14], and Saxena et al. in [15] and Perkovic et al. in [25] present device pairing methods based on device equipped with LEDs and a video camera as the receiver. In [16] Perkovic et al. propose a solution that allows an unaided user to initialize a relatively large number of wireless devices. The proposed solution is based on a multichannel Group message Authentication Protocol (GAP) in which information is transmitted over both a radio and Visible Light Channel (VLC).The main drawback of this solution is significant end-user involvement.

Li et al. in [17] propose a lightweight scheme for secure sensor association and key management in Wireless Body Area Networks (WBAN). However, their protocol is insecure in the attacker model where an adversary performs flipping attack in semi-authentic VLC. There are also plenty of other solutions which are adapted for use in WBANs and presented in [56], [57], [58]. The main drawback of these solutions is that they are applicable only to sensors that measure the same physiological signals and thus applicable only in WBANs. Shi et al. in [59] propose ASK-BAN, a lightweight fast authenticated secret key extraction scheme for intra-BAN communication. This scheme has a very low key generation rate compared with the other solutions.

7. CONCLUSION

In this work we designed and implemented two key distribution schemes for interface constrained wireless devices that are both secure, and extremely practical and easy to use. The proposed schemes have minimal hardware requirements on the wireless devices: one LED and one photodiode. Our schemes scale linearly in the number of wireless devices and enables a key distribution in a secure way in the presence of a very strong attacker.

First scheme comes in two flavours: (i) LIRA protocol which is based on secret key cryptography and does in fact require almost no computation from the device, and (ii) LIRA+ protocol which reduces the number of authentication messages transmitted as well as the computational cost on each network device compared to the original LIRA protocol. In both our solutions we use one-way visible light channel (VLC) of multi-touch screens (flashing displays) to initialize devices in a secure, usable and scalable way. Both protocols are shown to be secure in the very strong attacker model. We implemented LIRA protocol on commercially available platform and demonstrated through the extensive usability study that our solution has a good performance, scales linearly with the number of devices in the network and is easy to use.

In the second scheme we extended the attacker model to an extremely strong adversary who can observe the electromagnetic radiation from the screen and from the connecting cable between the screen and the video card. This scheme uses public-key cryptography and I-codes, a physical layer security primitive, for which transmission is synchronized by SYN signals over the visible light channel. We showed through detailed analysis that this construction is highly secure even when facing a very strong attacker who has a full control of the radio channel and to a great extent of the VLC channel too. We also showed through experiments and analytically that our public-key scheme is resistant to high non-malicious interference from other signals in the same part of the radio spectrum.

In the future, we will plan to work on more efficient coding techniques for the VLC channel. Likewise we will look at extending our solutions to enable efficient authentication and authorization of devices over public networks such as public telecom networks and Internet.

REFERENCES

1. Singh UR, Roy S, Mutum H. A survey on wireless sensor network security and its countermeasures: An overview. *International Journal of Engineering Science Invention* 2013; **2**.
2. Latré B, Braem B, Moerman I, Blondia C, Demeester P. A survey on wireless body area networks. *Wireless Networks* 2011; **17**(1):1–18.
3. Inc E. Estimote Beacons Real world context for your apps. `http://estimote.com/` 2014. [Online; accessed 25-July-2014].
4. Stackoverflow. Pairing iBeacon silently. `http://stackoverflow.com/questions/20077318/pairing-ibeacon-silently`. [Online; accessed 25-July-2014].
5. arstechnica. Crypto weakness in smart LED lightbulbs exposes Wi-Fi passwords. `http://arstechnica.com/security/2014/07/` 2014. [Online; accessed 25-July-2014].
6. Perrig A, Szewczyk R, Tygar J, Wen V, Culler DE. Spins: Security protocols for sensor networks. *Wireless networks* 2002; **8**(5).
7. Eschenauer L, Gligor VD. A key-management scheme for distributed sensor networks. *Proceedings of the 9th ACM conference on Computer and communications security*, ACM, 2002; 41–47.
8. Liu D, Ning P, Li R. Establishing pairwise keys in distributed sensor networks. *ACM Transactions on Information and System Security (TISSEC)* 2005; **8**(1):41–77.
9. Alliance Z. ZigBee Specification (Document 053474r06, Version 1.0). `http://www3.nd.edu/~mhaenggi/ee67011/zigbee.pdf/` June 2005. [Online; accessed 25-July-2014].
10. Kuo C, Luk M, Negi R, Perrig A. Message-in-a-bottle: user-friendly and secure key deployment for sensor nodes. *international conference on Embedded networked sensor systems*, ACM, 2007.
11. Law YW, Moniava G, Gong Z, Hartel P, Palaniswami M. Kalwen: A new practical and interoperable key management scheme for body sensor networks. *Security and Communication Networks* 2011; **4**(11).
12. Wong FL, Stajano F. Multichannel security protocols. *Pervasive Computing, IEEE* 2007; **6**(4).
13. Saxena N, Uddin MB. Automated device pairing for asymmetric pairing scenarios. *Information and Communications Security*. Springer, 2008.
14. Saxena N, Uddin MB. Blink'em all: Scalable, user-friendly and secure initialization of wireless sensor nodes. *Cryptology and Network Security*. Springer, 2009.
15. Saxena N, Uddin MB, Voris J. Universal device pairing using an auxiliary device. *Symposium on Usable privacy and security*, 2008.
16. Perkovic T, Cagalj M, Mastelic T, Saxena N, Begusic D. Secure initialization of multiple constrained wireless devices for an unaided user. *Mobile Computing, IEEE Transactions on* 2012; **11**(2).
17. Li M, Yu S, Lou W, Ren K. Group device pairing based secure sensor association and key management for body area networks. *INFOCOM, 2010 Proceedings IEEE*, IEEE, 2010; 1–9.
18. Chen CHO, Chen CW, Kuo C, Lai YH, McCune JM, Studer A, Perrig A, Yang BY, Wu TC. GAnGS: Gather, Authenticate 'n Group Securely. *Proceedings of the 14th Annual International Conference on Mobile Computing and Networking, MOBICOM*, MobiCom '08, ACM, 2008; 92–103.
19. Nithyanand R, Saxena N, Tsudik G, Uzun E. Groupthink: Usability of Secure Group Association for Wireless Devices. *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, Ubicomp '10, ACM, 2010; 331–340.
20. Mica2 specifications. `http://www.eol.ucar.edu/isf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf`. [Online;

accessed 25-July-2014].

21. Kovacevic T, Perkovic T, Cagalj M. Lira: A new key deployment scheme for wireless body area networks. *Software, Telecommunications and Computer Networks (SoftCOM), 2013 21st International Conference on*, IEEE, 2013.
22. electricimp. `http://electricimp.com/`. [Online; accessed 25-July-2014].
23. Kuhn MG. Electromagnetic Eavesdropping Risks of Flat-panel Displays. *Proceedings of the 4th International Conference on Privacy Enhancing Technologies*, PET'04, Springer-Verlag, 2004; 88–107.
24. Capkun S, Cagalj M, Rengaswamy R, Tsigkogiannis I, Hubaux JP, Srivastava M. Integrity codes: Message integrity protection and authentication over insecure channels. *Dependable and Secure Computing, IEEE Transactions on* 2008; **5**(4).
25. Perković T, Stančić I, Mališa L, Čagalj M. Multichannel protocols for user-friendly and scalable initialization of sensor networks. *Security and Privacy in Communication Networks*. Springer, 2009.
26. Gollakota S, Ahmed N, Zeldovich N, Katabi D. Secure in-band wireless pairing. *USENIX security symposium*, 2011.
27. Hou Y, Li M, Guttman JD. Chorus: Scalable In-band Trust Establishment for Multiple Constrained Devices over the Insecure Wireless Channel. *ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec, 2013.
28. Li M, Yu S, Guttman JD, Lou W, Ren K. Secure ad hoc trust initialization and key management in wireless body area networks. *ACM Trans. Sen. Netw.* Apr 2013; **9**(2).
29. Boyd C, Mathuria A. *Protocols for Authentication and Key Establishment*. Springer, 2003.
30. Cremers C. The scyther tool. a tool for the automatic verification of security protocols. `http://www.cs.ox.ac.uk/people/cas.cremers/scyther`. [Online; accessed 20-October-2014].
31. Basin DA, Cremers C, Meier S. Provably repairing the iso/iec 9798 standard for entity authentication. *Journal of Computer Security* 2013; :817–846.
32. Lowe G. A hierarchy of authentication specifications. IEEE Computer Society Press, 1997; 31–43.
33. Pasini S, Vaudenay S. Sas-based authenticated key agreement. *Public Key Cryptography-PKC*. Springer, 2006.
34. Wong FL, Stajano F. Multi-channel protocols for group key agreement in arbitrary topologies. *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, IEEE, 2006.
35. Cagalj M, Capkun S, Hubaux JP. Key agreement in peer-to-peer wireless networks. *Proceedings of the IEEE* 2006; **94**(2).
36. McCune JM, Perrig A, Reiter MK. Seeing-is-believing: Using camera phones for human-verifiable authentication. *Security and privacy, 2005 IEEE symposium on*, IEEE, 2005.
37. Choi JI, Jain M, Srinivasan K, Levis P, Katti S. Achieving Single Channel, Full Duplex Wireless Communication. *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom, 2010.
38. Pöpper C, Tippenhauer NO, Danev B, Capkun S. Investigation of Signal and Message Manipulations on the Wireless Channel. *Proceedings of the 16th European Conference on Research in Computer Security*, ESORICS, 2011.
39. Proakis JG. *Digital Communications*. McGraw-Hill, 1995.
40. Brooke J. SUS: A Quick and Dirty Usability Scale. *Usability Evaluation in Industry*, vol. 189, Taylor and Francis, 1996.
41. Zhu S, Setia S, Jajodia S. Leap: Efficient security mechanisms for large-scale distributed sensor networks. *Proceedings of the 10th ACM Conference on Computer and Communications Security*, CCS, 2003.
42. Karlof C, Sastry N, Wagner D. Tinysec: a link layer security architecture for wireless sensor networks. *Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004.
43. Anderson R, Chan H, Perrig A. Key infection: Smart trust for smart dust. *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on*, IEEE, 2004; 206–215.
44. Paek KJ, Kim J, Hwang CS, Song US. An energy-efficient key management protocol for large-scale

wireless sensor networks. *Multimedia and Ubiquitous Engineering, 2007. MUE'07. International Conference on*, IEEE, 2007.

45. Castelluccia C, Mutaf P. Shake them up! *Usenix Mobisys, 2005* ; .
46. Holmquist LE, Mattern F, Schiele B, Alahuhta P, Beigl M, Gellersen HW. Smart-its friends: A technique for users to easily establish connections between smart artefacts. *Ubicomp: Ubiquitous Computing*, Springer, 2001.
47. Lester J, Hannaford B, Borriello G. are you with me?–using accelerometers to determine if two devices are carried by the same person. *Pervasive computing*. Springer, 2004.
48. Stajano F. The resurrecting ducklingwhat next? *Security Protocols*, Springer, 2001.
49. Stajano F, Anderson R. The resurrecting duckling: security issues for ubiquitous computing. *Computer* 2002; **35**(4).
50. Balfanz D, Smetters DK, Stewart P, Wong HC. Talking to strangers: Authentication in ad-hoc wireless networks. *NDSS*, 2002.
51. Gehrmann C, Mitchell CJ, Nyberg K. Manual authentication for wireless devices. *Cryptobytes* 2004; **4**.
52. Vaudenay S. Secure communications over insecure channels based on short authenticated strings. *Advances in cryptology–CRYPTO 2005*, Springer, 2005.
53. Laur S, Nyberg K. Efficient mutual data authentication using manually authenticated strings. *Cryptology and Network Security*. Springer, 2006.
54. Wong FL, Stajano F. Multi-channel protocols. *Security Protocols*, Springer, 2007.
55. Soriente C, Tsudik G, Uzun E. Hapadep: human-assisted pure audio device pairing. *Information Security*. Springer, 2008.
56. Cherukuri S, Venkatasubramanian KK, Gupta SK. Biosec: A biometric based approach for securing communication in wireless networks of biosensors implanted in the human body. *International Conference on Parallel Processing Workshops*, IEEE, 2003.
57. Li FMLJY, Zhang Y. Aes based biometrics security solution for body area sensor networks 2009; .
58. Mana M, Feham M, Bensaber BA. Sekes (secure and efficient key exchange scheme for wireless body area network). *IJCSNS International Journal of Computer Science and Network Security* 2009; **9**(11).
59. Shi L, Yuan J, Yu S, Li M. Ask-ban: Authenticated secret key extraction utilizing channel characteristics for body area networks. *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, ACM, 2013.
60. Basin D, Cremers C, Meadows C. *Model Checking Security Protocols*. Springer, 2011. To appear.

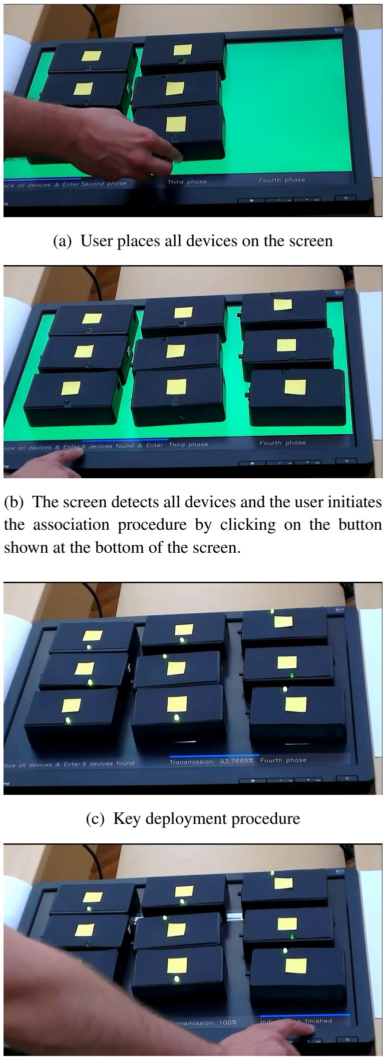
A. FORMAL VERIFICATION OF LIRA PROTOCOL

We have modelled LIRA and LIRA+ protocols in security protocol description language (SPDL) using Sycther tool as shown bellow. In our protocol definition we included role I and role C that represent the device S_i and the coordinator C, respectively. The roles definitions are sequence of all events used in our protocol: declarations, send, receive and claim events. Claim events are used to model intended security properties. Scyther is based on the operational semantics found in [60], and can be used in three ways: (i) to verify protocol, whether the security claims in the protocol description correct or not (verified claims); (ii) to analyse the protocol by performing complete characterization of the roles (characterized claims); (iii) to verify automatic claims by generating appropriate security claims for a protocol and verify them. The implementations of LIRA and LIRA+ protocols in Scyther are given bellow.

```
/* Scyther: v1.1.2.
 * Modeled from LIRA solution
 * Modeler: Tonko, april 2014
 * MAC function is implemented using a
 * hash function H(m) MAC(Ki,H(m))
 * three-pass
 * mutual authentication protocol
 * C - the body control unit
 * I - the i-th device
 * NAi - nonce generated by C
 * NBi - nonce generated by i-th device Si
 */

hashfunction H;

protocol lira(C,I)
{
  role C
  {
    fresh NAi: Nonce;
    var NBi: Nonce;

    send_1(C,I,I, NAi);
    recv_2(I,C,NBi,H(NBi, NAi,C,k(I,C)));
    claim(C,Running,I,NAi,NBi);
    send_3(C,I,H(NAi, NBi,k(I,C)));

    claim(C,Commit,I,NAi,NBi);
    claim(C,Alive);
    claim(C,Weakagree);
    claim(C,Niagree);
    claim(C,Nisynch);
  }
  role I
  {
    fresh NBi: Nonce;
    var NAi: Nonce;

    recv_1(C,I,I,NAi);
    claim(I,Running,C,NBi, NAi);
    send_2(I,C,NBi,H(NBi, NAi,C,k(I,C)));
    recv_3(C,I,H(NAi, NBi,k(I,C)));

    claim(I,Commit,C,NBi,NAi);
    claim(I,Alive);
    claim(I,Weakagree);
    claim(I,Niagree);
    claim(I,Nisynch);
  }
}
```

```
/* Scyther: v1.1.2.
 * Modeled from LIRA+ protocol
 * Modeler: Tonko, april 2014
 * MAC function is implemented using a
 * hash function H(m) MAC(Ki,H(m))
 * two-pass
 * mutual authentication protocol
 * C - the coordinator
 * I - the i-th device
 */

hashfunction H;

protocol liraplus(C,I)
{
  role C
  {
    claim(C,Running,C,I);
    send_1(C,I,I,H(I,C,k(I,C)));
    recv_2(I,C,H(I,k(I,C)));

    claim(C,Commit,C,I);
    claim(C,Alive);
    claim(C,Weakagree);
    claim(C,Niagree);
    claim(C,Nisynch);
  }

  role I
  {
    recv_1(C,I,I,H(I,C,k(I,C)));
    claim(I,Running,I,C);
    send_2(I,C,H(I,k(I,C)));

    claim(I,Commit,I,C);
    claim(I,Alive);
    claim(I,Weakagree);
    claim(I,Niagree);
    claim(I,Nisynch);

  }
}
```

DOI: 10.1002/sec
Prepared using secauth.cls



(a) User places all devices on the screen

(b) The screen detects all devices and the user initiates the association procedure by clicking on the button shown at the bottom of the screen.

(c) Key deployment procedure

(d) Upon a successful initialization all the devices should turn their green LED ON.

Figure 12. A user bootstraps a network of 9 sensing devices with LIRA protocol.

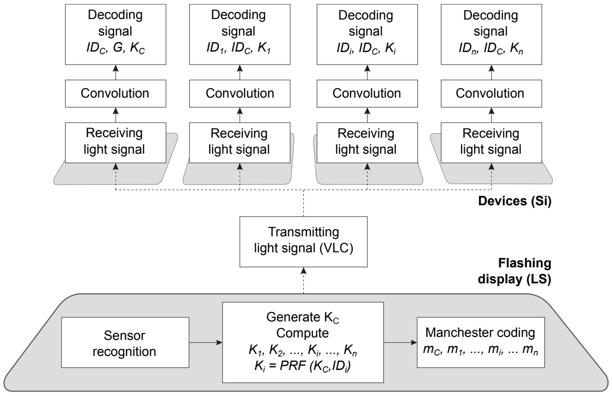

Decoding signal ID_C, G, K_C
Decoding signal ID_1, ID_C, K_1
Decoding signal ID_i, ID_C, K_i
Decoding signal ID_n, ID_C, K_n
Convolution
Convolution
Convolution
Convolution
Receiving light signal
Receiving light signal
Receiving light signal
Receiving light signal
Devices (Si)
Transmitting light signal (VLC)
Flashing display (LS)
Sensor recognition
Generate K_C Compute $K_1, K_2, ..., K_i, ..., K_n$ $K_i = PRF(K_C, ID_i)$
Manchester coding $m_C, m_1, ..., m_i, ... m_n$

Figure 13. Key generation procedure, its transmission over the Visible Light Channel, and decoding at the reception side.